PKI AND DIGITAL SIGNATURES PART 2

With the help of Peter Gutmann

Qualified Certificates

- Semi-technical name of public key certs that are
  - Issued to physical persons
    - As opposed to legal persons, devices etc.
  - Issued after physically checking identity
- Can also be issued to pseudonym
  - Must be traceable to real name
- Implication: digital signature applications should use qualified certs

S/MIME

- Defines format for attaching signature to document
- Used for signed and encrypted E-mail
- RFC 3851
- Uses ASN.1 syntax

XML Signature

- Signature format using XML syntax
- Fairly complex machinery
  - 73 pages vs. 36 pages of S/MIME
- Problem: XML data can be encoded in different ways
  - Use of whitespace
  - Ordering of attributes
  - ...

XML Signature (2)

- Canonicalization
  - Defines unique way of encoding XML content
  - Similar to DER encoding in ASN.1

ETSI

- European Telecommunications Standards Institute
- Issues standards compliant with the European Commission e-sign directive (1999/93/EC)
- Qualified certificate profile
- Policy requirements for CAs issuing qualified certificates
- Time-stamping profile

ETSI Signature Formats

- CMS Advanced Electronic Signatures (CAdES)
- XML Advanced Electronic Signatures (XAdES)
- PDF Advanced Electronic Signature (PAdES)

- Define signature formats with different amounts of proof info

ETSI Signature Formats (2)

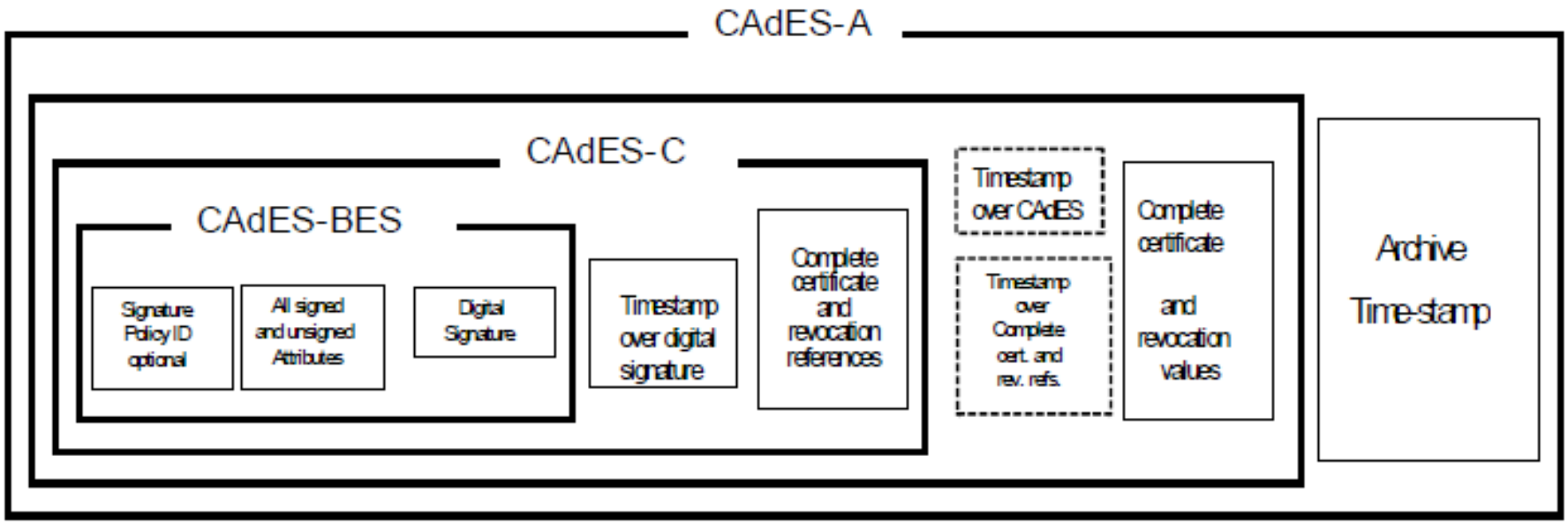
- Basic Electronic Signature
  - Just data + signature
- Explicit Policy-based Electronic Signature
  - BES + signature policy ID
- Electronic Signature with Time
  - BES/EPES + time stamp
- ES with Complete Validation Data References
  - CAdES-T + references to all used certs and revocation info (CRLs, OCSP responses)

ETSI Signature Formats (3)

- EXtended Long Electronic Signature
  - CAdES-C + all the cert and revocation info (contents)
- EXtended Long Electronic Signature with Time
  - XAdES-X Long + time stamp over CAdES-C
  - XAdES-X Long + time stamp over cert and revocation references

ETSI Signature Formats (4)

- Archival Electronic Signature
  - XAdES-X Long type 1 or 2 + time stamp over all the data



Pretty Good Privacy

- a.k.a PGP
- PKI for sending secure E-mail
- Two people who know each other can exchange keys
  - Self-signed certificates
- If A knows B, A can sign B's key and E-mail address
  - Effectively a certificate
  - If C trusts A, he ca accept A's Certificates

Web of Trust

- You can mark some people as trusted introducers
- You can specify how the trust decreases over several steps
- „Six degrees of separation”
- Certificates are stored on key servers
  - Revocation is fast and simple

Pretty Good Privacy (2)

- Certificates are key-based
  - One certificate can contain several E-mail addresses
- Authentication keys are used to certify confidentiality keys
  - Confidentiality keys can have much shorter life cycle
- Works well and fits the purpose

SPKI

- Simple Public Key Infrastructure
- Names are useless
- Certificates bind key to authorization or capability
- Uses local names
  - Locally meaningful
  - Locally unique
- Authorization can require *m* out of *n* consensus

SPKI (2)

- Use S-expressions instead of ASN.1 or XML
- (name fred)
  - Owner's name is fred
- (name (hash sha1 |TLCgPLFlGTzyUbcaYLW8kGTEnUk=|))
  - Globally unique key ID
- (ftp (host ftp.warez.org) (dir /pub/warez))
  - Keyholder can access FTP site

SPKI (3)

- All names, authorizations and capabilities are local
  - Must make sense in given application
- SPKI is all about delegation
  - Can delegate all or part of access
  - Can restrict re-delegation
- If you want to trust other service providers, just sign their keys
  - Trust anchor is your own key

SPKI (4)

- Example authorization chain
  - A may access resource X. Signed: Service Provider
  - B may access resource X. Signed: A
  - Service provider, please allow me to access X. Signed: B
- Verification
  - Service provider checks signatures from B -> A -> own key
  - No need for external trusted party

SPKI (5)

- Internally, SPKI certificates are represented as 5-tuples
  - <Issuer, Subject, Delegation, Authority, Validity>
  - Issuer, subject are identified via the global key ID
  - Delegation = Subject can delegate authority
  - Authority = Authority granted to the certificate subject
  - Validity = Validity period and/or online validation test information

SPKI (6)

- Certificates are processed using general-purpose tuple reduction algorithm
- <I1, S1, D1, A1, V1> + <I2, S2, D2, A2, V2> = <I1, S2, D2, A12, V12>
  - if S1 = I2 and D1 = true
  - A12 = intersection(A1, A2)
  - V12 = intersection(V1, V2)
- Eventually some chains of authorization statements will reduce to
  <Trusted Issuer, x, D, A, V>

SPKI in Practice

- Good idea
- Simple implementation, working solution
- Addresses real problem
- Oriented towards machine processing

- **Not used in practice**
  - Main reason: not X.509

Certificates and Signatures in Real Life

Trouble with Identity

- „I will pay back 100EUR by Monday, signed by X”
  - So?
  - Who's X?
  - How do I know he will pay back?
- People operate on different roles
  - Margus Freudenthal as citizen of Estonia
  - Margus Freudenthal as member of board of the Tallinn go club

Long-Term Validation

- Set the date on your computer to 2015 and try to verify .sdoc file

- Current mechanisms do not provide long-term security
- It is in principle possible to solve this
  - Technical complexity is high
  - Non-standard
  - People do not understand the problem

Technical Non-Repudiation

- There is no direct connection between the signer and the signature
- You just push some button and something happens
  - „Software did not make me aware of consequences of my actions”
  - „The virus did it!”
  - „Oops, I published the private key...”
  - „I discovered later that the key was stolen”

Technical Non-Repudiation (2)

- You can repudiate handwritten signatures
  - You can testify that you were under duress
- If legislation forbids repudiation, then it is too harsh and makes unreasonable demands
- What are the actual guarantees that the relying parties have?
- Certificate suspension destroys non-repudiation

Technical Non-Repudiation (3)

- There are no legal precedents for disputes about validity of an electronic signature
  - Nobody knows how the process will work and what could be the decision
- Estonian digital signatures operate based on good faith
  - No one knows what would happen if somebody would deny a signature

Digital Signature Legislation

- Prescriptive
  - You must do X to comply
  - Germany, Sweden
- Hands-off
  - Anything reasonable is fine
  - California

Digital Signature Legislation

- Estonia has hybrid approach
  - General requirements for digital signature mechanism
  - Specific mechanisms and rules
    - Certification authorities
    - Time-stamping authorities

Interoperability

- Very low quality of standards
- Complexity
  - Cert verification algorithm is 30 pages
  - Path building and discovery: separate 78 page specification
  - Part of it stems from long history of design by committee
- Spread of profiles

Interoperability (2)

- Very vague specifications
  - Everything left as a policy decision
  - Software cannot make binary decision
  - Cannot use for automatic processing of low-value transactions
- Very low quality of software
  - Almost nobody does all the checks because stuff will not work

No Real Business Need

- CA is additional party
  - Wants money
  - Must be trusted
  - Provides identity of a person which is not very useful
  - Does not offer any guarantees
- Better to just exchange keys and do business

Certificates in Browsers

- Browsers come preinstalled with hundreds of certificates
  - All of them are trusted
  - Takes superhuman effort to disable them
- Selling SSL server certs is a racket
  - „Pay us money so your visitors will not get confusing warning messages”

Developing Closed Systems

- You can ignore the nonsense part
- Make concrete rules
  - Who does what
  - Who is responsible
  - How disputes are handled
  - What happens if something goes wrong
- Freeze the technical specifications
- Use only part that is useful to you
  - X.509 as public key container format

Developing Closed Systems (2)

- Do end-users need public keys?
  - Server-based signatures
  - Ask bank to authorize transaction instead of verifying certificate
  - Managing small amount of public keys of stable entities is more realistic
    - Also possibly more secure

Closed System Examples

- Estonian X-Road
  - Does not interface with the outside world
  - Clear trust relationships
  - Uses X.509 certs in very specific, well-defined way
- Estonian ID card PKI (partially)
  - One CA
  - One digital signature software
  - Specifications with fair quality